Coloração de Mapas com MVNS

Felipe Pinheiro Feliciano
Universidade Estadual do Rio de Janeiro
IPRJ / Engenharia da Computação
felipe.feliciano@grad.iprj.uerj.br

Pedro Muniz Vargas
Universidade Estadual do Rio de Janeiro
IPRJ / Engenharia da Computação
pedro.vargas@grad.iprj.uerj.br

RESUMO

Dado um grafo G = (V, E) composto por um conjunto de vértices V e um conjunto de arestas E, problema da Coloração de Grafos (PCG) consiste em atribuir uma cor a cada vértice do grafo, de modo que vértices adjacentes não possuam a mesma cor, utilizando o menor número de cores possível. Neste trabalho procuramos incorporar o uso de memória ao VNS básico, o tornando MVNS, na tentativa de contribuir com resultados médios mais promissores para o PCG.

ABSTRACT

Given a graph G = (V, E) composed of a set of vertices V and a set of edges E, the Graph Coloring Problem (GCP) consists of assigning a color to each vertex of the graph, in such a way that adjacent vertices do not share the same color, using the minimum number of colors possible. In this work, we aim to incorporate the use of memory into the basic VNS, making it MVNS, in an attempt to contribute to more promising average results for the GCP.

1. Introdução

Na teoria dos grafos, a coloração de grafos é uma abordagem especializada da rotulagem de grafos, onde os elementos do grafo recebem rótulos denominados "cores" de acordo com determinadas restrições. Em sua forma mais simples, a coloração de vértices envolve a atribuição de cores aos vértices do grafo, garantindo que vértices adjacentes não compartilhem a mesma cor. De maneira análoga, a coloração de arestas envolve atribuir cores às arestas de modo que arestas adjacentes não tenham a mesma cor, enquanto a coloração de faces de um grafo planar envolve atribuir cores a cada face ou região, garantindo que faces adjacentes não compartilhem uma fronteira com a mesma cor.

A coloração de vértices é fundamental para esse tema, sendo que outros problemas de coloração podem ser transformados em versões de coloração de vértices. Por exemplo, a coloração de arestas de um grafo é equivalente à coloração de vértices de seu grafo linha, e a coloração de face de um grafo planar é uma coloração de vértices do seu dual planar. No entanto, alguns problemas de coloração que não envolvem vértices são frequentemente estudados separadamente devido a perspectivas específicas e à natureza dos problemas, como é o caso da coloração de arestas.

A prática de usar cores tem origem na coloração de países em um mapa, onde cada face é literalmente colorida. Essa prática foi generalizada para a coloração de faces de um grafo incorporado no plano. A partir da dualidade planar, a coloração passou a envolver os vértices, tornando-se uma generalização para todos os grafos. Em representações matemáticas e na informática, é comum usar os primeiros números inteiros positivos ou não negativos como "cores", embora, em geral, qualquer conjunto finito possa ser utilizado como o "conjunto de cores". *A natureza do problema de coloração depende do número de cores, mas não das cores em si.*

A coloração de grafos tem diversas aplicações práticas e desafios teóricos. Além dos problemas clássicos, diferentes restrições podem ser aplicadas ao grafo, à forma como uma cor é atribuída ou até mesmo à própria cor. A coloração de grafos alcançou popularidade entre o público em geral por meio de jogos de quebra-cabeça, como o Sudoku. Atualmente, a coloração de grafos permanece como um campo de pesquisa ativo.

O uso de memória em meta-heurísticas incorpora parte do modelo de multi-memória proposto em 1968 por Atkinson e Shiffrin (1968), baseado na hipótese de como este processo deve funcionar nos seres humanos, poderia ser adaptado para um programa. Consideram-se dois tipos de memória:

• **Memória de curto prazo**: Utilizada quando necessitamos de certas informações apenas em caráter imediato.

• **Memória de longo prazo**: Existem informações que guardamos na memória, podendo recordá-las depois de décadas. Nesta proposta, iremos incorporar ao VNS básico o uso de análogos computacionais dessas memórias, na tentativa de contribuir com resultados médios mais promissores para o PCG.

1.1. Objetivos

O principal objetivo do trabalho é estudar e desenvolver uma forma de solucionar o problema, aplicado ao PCG. Como objetivos específicos, cita-se:

• Realizar um levantamento bibliográfico sobre o PCG;
• Propor e desenvolver uma forma de aplicar o MVNS na Coloração de Grafos;
• Analisar a viabilidade e o comportamento do algoritmo em comparação com outras metaheurísticas;

2. O Problema

A coloração de grafos teve sua origem em 1852 quando o estudante Francis Guthrie propôs o famoso problema das quatro cores. Nesse desafio, ele questionou se seria possível colorir qualquer mapa usando apenas quatro cores, de modo que países com fronteiras em comum não compartilhassem a mesma cor. Em 1879, o matemático Kempe apresentou uma solução inicial para esse problema, demonstrando o teorema das quatro cores. Posteriormente, em 1880, o físico matemático Peter Guthrie Tait começou a explorar a coloração das arestas como uma abordagem para resolver o mesmo problema, oferecendo outra demonstração para o teorema. No entanto, em 1890, Percy John Heawood apontou erros na demonstração de Kempe, e em 1891, Julius Peter Christian Petersen também indicou falhas na demonstração de Tait.

Ao longo dos anos, diversos estudos e contribuições foram feitos para resolver o problema proposto por Guthrie. Em 1976, os matemáticos Hapel e Akken resolveram o problema utilizando computadores em sua demonstração, gerando debates na comunidade matemática. Eles definiram 1936 configurações a serem verificadas por computador, utilizando aproximadamente 1200 horas de computação. Assim, foi demonstrado que qualquer grafo planar, ou seja, um grafo que pode ser desenhado no plano sem que suas arestas se cruzem, é 4-colorível. No entanto, o problema da coloração de grafos vai além disso, buscando encontrar o número mínimo de cores necessário para colorir qualquer grafo. O problema da coloração de grafos é proeminente na teoria dos grafos, sendo um dos problemas mais estudados na comunidade acadêmica. Classificado como NP-difícil, ainda não existe um algoritmo conhecido que forneça uma resposta exata para ele em tempo polinomial. Portanto, várias técnicas são empregadas para encontrar a melhor solução possível para esse desafio.

Dentro da comunidade científica de Computação e Pesquisa Operacional, o problema da coloração de grafos desperta grande interesse. Além disso, diversos estudos estabelecem conexões entre a coloração de grafos e problemas do cotidiano, como agendamento de horários educacionais, planejamento de projetos, alocação de registradores, atribuições de canais de redes sem fio, agendamento de tarefas em máquinas, agendamento de competições esportivas, planejamento de tráfego aéreo, entre outras aplicações práticas. Essa relação direta com problemas do mundo real destaca ainda mais a relevância e a aplicabilidade da coloração de grafos em diversas áreas.

2.1 O VNS

A Meta-heurística VNS, ou Variable Neighborhood Search (Busca em Vizinhança Variável), é um método de otimização global que pertence à classe de algoritmos de busca local. Foi proposta por Mladenović e Hansen em 1997 como uma extensão da busca em vizinhança fixa. A ideia fundamental por trás é explorar diferentes estruturas de vizinhança em torno de uma solução corrente, permitindo que o algoritmo escape de mínimos locais e busque soluções de melhor qualidade.

O algoritmo começa com uma solução inicial, define um conjunto de vizinhanças, e realiza buscas locais em cada vizinhança. Após um número específico de iterações, uma perturbação é introduzida, alterando a solução corrente. Se a nova solução for melhor, ela é aceita; caso contrário, é descartada. Esse processo de busca em vizinhança variável é repetido até que um critério de parada seja atendido. A VNS é eficaz na exploração do espaço de solução, permitindo escapar de mínimos locais e é aplicada com sucesso em problemas de otimização combinatória, adaptando dinamicamente as estruturas de vizinhança durante a busca.

A VNS é chamada de "Busca em Vizinhança Variável" porque, ao longo do processo, ela ajusta dinamicamente as estruturas de vizinhança utilizadas. Isso ajuda a evitar a convergência prematura para mínimos locais e permite uma exploração mais ampla do espaço de solução.

Essa meta-heurística tem sido aplicada com sucesso em diversos problemas de otimização combinatória, como o Problema do Caixeiro Viajante, o Problema da Mochila, entre outros. No entanto, é importante ajustar os parâmetros da VNS de acordo com as características específicas do problema em questão para obter os melhores resultados.

2.2 Memória

Podemos considerar duas variações de controle de memória existentes:

• Controle por vértice (vertex control) – se um vértice foi usado, sua posição não poderá ser trocada durante uma ou mais iterações;

• Controle por movimento (movement control) – se um movimento de troca é feito, este não pode ser utilizado durante uma ou mais iterações.

O controle de memória inserido no VNS básico pode ser feito por vértice ou por movimento, os detalhes sendo apresentados adiante. Esta inserção acontece em duas situações bem definidas:

• Memória da agitação feita na solução atual
Procura auxiliar na condução da geração das próximas soluções $(x \rightarrow x')$, (fase de agitação da vizinhança atual). Tal controle usa o mesmo princípio da meta-heurística Busca Tabu, evitando assim o efeito de ciclagem, que é a possibilidade de retorno a uma solução imediatamente anterior à atual;

• Memória da melhora da solução atual
O seu uso está associado apenas às melhoras ocorridas na fase de busca local $(f(x'') < f(x))$, ou seja, quando estas geram uma nova solução x, sendo o uso dessas informações preservado na geração das próximas vizinhanças. Isto equivale a um dos esquemas de cruzamento (crossover) da meta-heurística Algoritmos Genéticos, Goldberg (Go,89), onde se procura fazer com que as boas características dos pais sejam herdadas pelos filhos.

3. Comparação com a Literatura

A partir da tabela de resultados (apresentada mais abaixo) é possível notar que apesar desta meta-heurística não atingir o melhor resultado conhecido, ela é muito promissora pois, apresenta resultados muito parecidos com o GRASP proposto no artigo [17]. O fato de não atingir o melhor resultado pode ter sido causado pela falta de experiencia dos autores em implementações de meta-heurísticas, ou pela escolha infeliz de vizinhanças.

	Melhor Valor		MVNS	GRASP	mipHeuristic
Instancia	Número de cores	Tempo (s)	Número de cores	Número de cores	Número de cores
DSJC125.1	**5**	56	7	7	**5**
DSJC125.5	**17**	981	25	23	20
DSJC125.9	**44**	1800	57	51	47
DSJC250.1	**8**	316	13	12	9
DSJC250.5	**28**	635	42	41	36
DSJC500.1	**12**	4294	19	19	15
DSJC500.5	**48**	8580*	73	70	64
DSJC1000.1	**20**	8580**	31	30	26

* rodou apenas 6 iterações
** rodou apenas 4 iterações

Ao longo deste estudo, conduzimos uma série de testes com a finalidade de alcançar o desempenho mais otimizado possível para as instâncias especificadas na tabela acima. Ao analisarmos os resultados provenientes da mipHeuristic e do GRASP, buscamos identificar as tendências e padrões que poderiam contribuir para aprimorar o desempenho global do sistema. Além disso, é possível notar que este é um problema muito complexo, haja vista que nenhuma das meta-heurísticas analisadas como referencia atingiu o melhor valor em 100% dos casos. Porém, todas as 3 apresentaram resultados satisfatórios em relação ao número de cores, visto que os grafos utilizados tem tamanhos de 125, 250, 500 e 1000 vértices.

Considerando o contexto do desenvolvimento deste estudo, incluindo as instâncias empregadas nos testes, os tempos pré-definidos e outros fatores específicos, observa-se que a heurística apresenta resultados competitivos em comparação com outros métodos. Vale ressaltar que esse desempenho favorável não implica que a heurística seja limitada a grafos menores, indicando sua viabilidade para contextos mais amplos.

Embora a comparação com o estado da arte não tenha superado os resultados previamente alcançados por outros pesquisadores, é crucial salientar que isso não descarta a utilidade da heurística proposta neste artigo. Pelo contrário, indica que há espaço para a realização de estudos mais aprofundados nessa área, explorando potenciais melhorias e adaptações da heurística para alcançar resultados ainda mais robustos.

```python
while True:
        node = random.choice(list(new_sol.keys()))
        if node not in self.memory:
        break
```

Este trecho do código é importante de ser ressaltado pois realiza o controle de vértices, ou seja, faz a memoria do algoritmo. Isso é feito escolhendo um vértice aleatoriamente enquanto este não esteja na lista de tabu.

4. Conclusão

Este estudo propôs a aplicação de uma heurística, a MVNS, no contexto do Problema da Coloração de Grafos. Dada a relevância substancial desse problema nas esferas científicas de computação e matemática, foi imperativo compreender profundamente a sua importância, aplicações e explorar parte dos estudos preexistentes para proporcionar aprimoramentos significativos. Um extenso levantamento bibliográfico foi conduzido, abrangendo os trabalhos mais notáveis relacionados ao tema e os algoritmos de destaque. Além disso, o problema foi programado e a heurística proposta foi implementada, culminando em uma análise comparativa com outras heurísticas da literatura, com o adendo que devido a escolha de vizinhanças, os resultados são promissores mesmo não tendo atingido ótimo em todos os casos.

Os resultados revelaram que, apesar de os primeiros registros da aplicação dessa estratégia serem recentes, e considerando a abundância de estudos ainda por realizar, essa abordagem exibiu resultados preliminares promissores. Esses resultados não apenas consolidam a aplicação da MVNS como uma alternativa viável, mas também indicam seu potencial de aprimoramento em trabalhos futuros.

No contexto de perspectivas futuras, há diversas direções promissoras a serem exploradas. A análise holística dos elementos apresentados sugere que a meta-heurística empregada para o Problema da Coloração de Grafos não apenas oferece resultados satisfatórios, mas também abre portas para um crescimento significativo dentro da área. Diante da magnitude do problema em questão, antevê-se uma trajetória ascendente para a relevância e eficácia dessa abordagem em futuras investigações e aplicações práticas.

5. Referências

[1] J. L. Gonzáles-Velarde e M. Laguna, Tabu Search with Simple Ejection Chains for Coloring Graphs, Annals of Operations Research, 117, 165-174, 2002

[2] M. Laguna e R. Martí, A GRASP for coloring sparse graphs, Computational Optimization and Applications, 19(2), 165-178, 2001.

[3] M. Chams, A. Hertz e D. de Werra, Some experiments with simulated annealing for coloring graphs, European Journal of Operation Research 32, 260-266, 1987.

[4] C. Fleurent e J.A. Ferland, Genetic and Hybrid Algorithms for Graph Coloring, Annals of

Operation Research, 63, 437-464, 1996.

[5] M. Trick, “Graph Coloring Instances”, disponível em
http://mat.gsia.cmu.edu/COLOR/instances.html

[6] FM. R. Garey e D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co., San Francisco, 1979

[7] F.T. Leighton, A graph coloring algorithm for large scheduling problems, J. Res. Nat. Bur. Standards, 84, 489-506, 1979.

[8] A. Hertz e D. de Werra, Using Tabu Search Techniques for Graph Coloring, Computing 39, 345-351, 1988.

[9] M. Chiarandini e T. Stützle, An application of Iterated Local Search to Graph Coloring, Proceedings of the Computational Symposium on Graph Coloring and its Generalizations (D.S. Johnson, A. Mehrotra e M. Trick, editores), 112-125, 2002.

[10] D. S. Johnson, C. A. Aragon, L. A. Mcgeoch e C. Schevon, Optimization by Simulated Annealing: An Experimental Evaluation – Part II (Graph Coloring and Number Partitioning), Operations Research, 31, 378-406, 1991.

[11] D. Brelaz, New methods to color the vertices of a graph. Communications of the ACM, 22(4), 251–256, 1979.

[12] F. Glover e M. Laguna, Tabu Search. Kluwer, Boston, 1997.

[13] P. Hansen, The steepest ascent mildest descent heuristic for combinatorial programming, Congress on Numerical Methods in Combinatorial Optimization, Capri, 1986.

[14] B. Selman, H. Levesque, e D. Mitchell, A new method for solving hard satisfability problems, Proceedings of the 10th National Conference on Artificial Intelligence, 1992.

[15] W. van Dorst, “BogoMips mini-Howto”, disponível em
http://www.tldp.org/HOWTO/BogoMips/

[16] Boaventura-Netto, “o uso de memória no vns básico aplicado ao problema quadrático de alocação”, disponível em http://www.din.uem.br/sbpo/sbpo2009/artigos/55490.pdf

[17] M. Rego, H. Santos, Algoritmos para o Problema de Coloração de Grafos, UFOP - Universidade Federal de Ouro Preto.